

## LRF010 参考设计简介

### 一、性能特点

频段： 2.405~2.480 , 一共 16 个信道

调制方式： DSSS

通讯速率： 250Kbps/2Mbps

工作电压： 3.3V +/-10%

睡眠电流： 3uA

发射功率： 10dBm ( 空旷距离 500 米 )

接收灵敏度： -100dBm

发射电流： 54mA

接收电流： 26mA

天线： 板上陶瓷天线 LTCC 或者 MMCX 外置天线接口

尺寸： 38.35mm \* 14.00mm

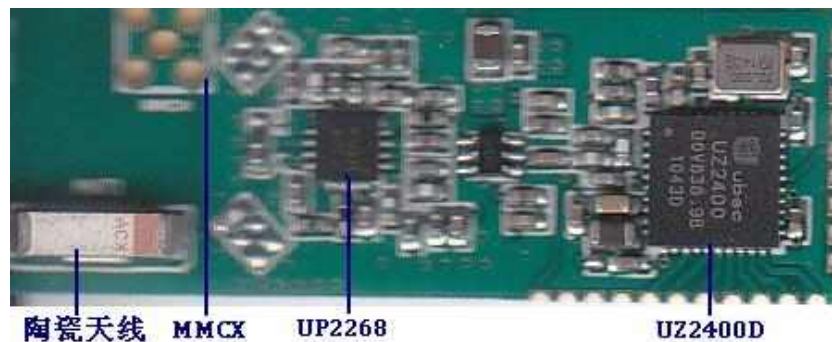
- LRF010 支持简单的星状网络。
- 发射接收缓冲区长度 128Bytes,Tx Ping-pong FIFOs。
- 高接收灵敏度-100dBm.
- 抗干扰能力突出。
- AES-128 数据加密功能。



- 支持 RSSI 读取。
- 支持防碰撞机制 ( CSMA/CA )。
- 自动应答机制 ( Auto-ACK )。
- 自动重发、硬件 CRC 校验以及自动丢掉错误数据包功能。
- 硬件自动识别数据包地址。

## 二、 模块介绍

如下图 ,LRF010 射频模块由 ACX 陶瓷天线、UBEC 的功放 UP2268 以及 UZ2400D 收发器组成。可直接焊接在母板上 , 供电 3.3V , 通过 SPI 口与 MCU 连接。具体 PIN 脚定义和结构图见 LRF010 中文说明书.Pdf 文档。



## 三、 硬件问题描述

### 1. 硬件连接

18 脚 ,     VCC 为 3.3V。

16 脚 ,     时钟输出脚 , 可给外部电路提供 4 组频率时钟。

8~11 脚 ,   SPI 口。

7 脚 ,       INT 中断脚 , 当 UZ2400D 有事件发生 , 产生信号 , 通知 MCU 处理

( 触发方式可设置 , 并可读取中断寄存器识别中断类型 )。不使用时 , 可接 10K 上拉电阻

6 脚 ,       外部唤醒功能 , 外部 MCU 可通过电平唤醒 UZ2400D 高低点平可

设置 ( 此管脚不使用时接 10K 上拉电阻 , 通过设置寄存器实现睡眠唤醒功能 )。

5 脚 ,       复位脚 , 低电平有效 , 外部 MCU 可通过此脚复位 UZ2400D 芯片。不使



接 10K 上拉电阻。

4 脚， GPIO0

## 2. 硬件调试

- A 上电时测量模块 16 脚是否有时钟输出，如有时钟输出，表示 LRF010 开始工作；
- B 在初始化 MCU 以及 SPI 后，写一短寄存器，然后读此寄存器，如果读出值等于写入值，UZ2400D SPI 口工作正常，此时硬件连接基本 OK。
- C 模块工作时，可通过电流大小简单判断是否工作正常：空闲模式 7mA，接收 20 多 mA，打开 PA 时发射 50 多 mA。
- D 通讯正常，如距离非常近几米，请检查 PA 是否打开，打开时模块上 UZ2400 GPIO

状态如下表

	GPIO1	GPIO2
TX	1	0
RX	0	1

## 四、 软件问题描述

- 1. **问题说明** 初始化函数以及功率设置请最新版本设置，具体见此文档的软件问题描述的第 7 项
- 2. **软件流程** 初始化 MCU→初始化 SPI→初始化 UZ2400→初始化模块地址  
→进入事件（发送数据或等待数据）。
- 3. **发送接收** UZ2400D 初始化、发送、接收函数实例请见  
UZ2400D Sample Code.pdf
- 4. **功放使用** 在接收发送时调用以下函数。

```
void UmEnablePA(){  
    spi_lw(TESTMODE, 0x29);           //Enable the GPIO control the Ex-P
```

automatically

```
spi_lw(RFCTL53, 0x00);           //Power Setting according to the LRF010 SPEC  
  
spi_lw(RFCTL74, 0xC5);  
  
spi_lw(RFCTRL3, 0x00);}
```

**5. 读写操作** 模块发送数据时，先将数据写入 TX FIFO，然后触发发送机，UZ2400D 自动发送数据；接收数据时，当 UZ2400D 收到数据时，产生中断给 MCU，MCU 从 RX BUFF 中读取数据；模块工作模式设置以及状态控制通过读取长、短寄存器实现。长、短寄存器和 TX/RX BUFF 地址机制各不相同，因此，通过 6 个函数实现读写操作，请使用时注意区分。

**6. 寄存器读写测试** 请测试短寄存器读写操作、长寄存器读写操作。写 TX FIFO 操作，通过长寄存器读函数读取 TX FIFO。

## 7. 初始化代码

```
void UmInit(){  
    unsigned char i;  
    do  
        spi_sw(GATE_CLK, 0x20);  
    while((spi_sr(GATE_CLK)&0x20)!=0x20);  
    spi_sw(PAONSETUP, 0x08);  
    spi_sw(FIFOEN, 0x94);  
    spi_sw(TXPEMISP, 0x95);  
    spi_sw(BBREG3, 0x50);  
    spi_sw(BBREG5, 0x07);  
    spi_sw(BBREG6, 0x40);  
    spi_lw(RFCTRL0, 0x03);  
    spi_lw(RFCTRL1, 0x02);  
    spi_lw(RFCTRL2, 0x66);  
    spi_lw(RFCTRL3, 0x00);  
    spi_lw(RFCTRL4, 0x06);  
    spi_lw(RFCTRL6, 0x30);  
    spi_lw(RFCTRL7, 0xec);  
    spi_lw(RFCTRL8, 0x8C);  
    spi_lw(GPIO_DIR, 0x00);  
    spi_lw(SEC_APP, 0x20);  
    spi_lw(RFCTL50, 0x05); // DC/DC off
```

```
spi_lw(RFCTL51,0xC0);
spi_lw(RFCTL52,0x01);
spi_lw(RFCTL53,0x00);
spi_lw(RFCTL59,0x00);
spi_lw(RFCTL73,0x40);    //DC-DC off
spi_lw(RFCTL74,0xC5);    //DC-DC off
spi_lw(RFCTL75,0x13);
spi_lw(RFCTL76,0x07);
spi_sw(INTMSK, 0x00);
spi_sw(SOFT_RST, 0x02);
spi_sw(RFCTL, 0x04);
spi_sw(RFCTL, 0x00);
spi_sw(RFCTL, 0x02);
i=0;while(--i);
spi_sw(RFCTL, 0x01);
i=0;while(--i);
spi_sw(RFCTL, 0x00);
spi_sw(SLPACK, 0x7f);
while(spi_lr (RFSTATE) != 0x50);}
```

**8. 模块地址** 由于应用时多用 Address Filter 模式，请注意发送模块、接收模块以及发送数据包数据里面地址设置。 实例说明如下：

下面分别为接收模块地址设置函数、发送模块函数以及发送数据包定义，

- ( 1 ) 处为网络 ID，发送、接收以及发送数据包设成相同值；
- ( 2 ) 处为接收模块地址，发送数据包中对应处设置与接收模块一致；
- ( 3 ) 处为接收模块 MAC 地址；
- ( 4 ) 处为发送模块地址，发送数据包中对应处设置与发射模块一致
- ( 5 ) 处为发送模块 MAC 地址；

```
void UMID_RX()
{
    spi_sw(PANIDL,0xaa);
    spi_sw(PANIDH,0xaa);    //网络 ID    ( 1 )
    spi_sw(SADRL,0x12);
```



```
spi_sw(SADRH,0x34);    //当前接收模块地址    ( 2 )
spi_sw(EADR0,0xA1);    //当前接收模块 MAC 地址    ( 3 )
spi_sw(EADR0,0xA2);
spi_sw(EADR0,0xA3);
spi_sw(EADR0,0xA4);
spi_sw(EADR0,0xA5);
spi_sw(EADR0,0xA6);
spi_sw(EADR0,0xA7);
spi_sw(EADR0,0xA8);

}    //接收模块

void UMID_TX()
{
    spi_sw(PANIDL,0xaa);
    spi_sw(PANIDH,0xaa);    //网络 ID    ( 1 )
    spi_sw(SADRL,0x11);
    spi_sw(SADRH,0x11);    //当前发送模块地址    ( 4 )
    spi_sw(EADR0,0x11);    //当期接收模块 MAC 地址    ( 5 )
    spi_sw(EADR0,0x22);
    spi_sw(EADR0,0x33);
    spi_sw(EADR0,0x44);
    spi_sw(EADR0,0x55);
    spi_sw(EADR0,0x66);
    spi_sw(EADR0,0x77);
    spi_sw(EADR0,0x88);

}    //发送模块

uchar data TXFIFO[14]=
{
    0x00,    //header length
    0x0c,    //frame length
    0x21,    //header loew byte
    0x88,    //header high byte
    0x00,    //sequence no.
    0xaa,    //接收模块地址 PAN ID low byte
    0xaa,    //接收模块地址 PAN ID high byte    ( 1 )
    0x12,    //接收模块地址 low byte
    0x34,    //接收模块地址 high byte    ( 2 )
    0xaa,    //发送模块地址 PAN ID low byte
    0xaa,    //发送模块地址 PAN ID high byte    ( 1 )
    0x11,    //发送模块地址 low byte
```





```
0x11,    //发送模块地址 high byte  
0x04  
}        //发送数据包
```

## 五、 资料列表

ITEM	NAME	README
模块资料	LRF010 中文资料.Pdf	LRF010 模块中文资料
芯片资料	UZ2400D.pdf	UZ2400D. SPEC
源代码	UZ2400D_MEG8.rar	UZ2400D_MEG8 Sample Code
	UZ2400D Sample Code.pdf	UZ2400D Sample Code
PCB 文档	RF module interface board layout guide.pdf	模块放置指导